

JS Level 0



Введение



Введение

Сегодняшнее наше занятие будет посвящено оформлению документов (вернее их частей) и обсуждению CSS.



CSS



CSS

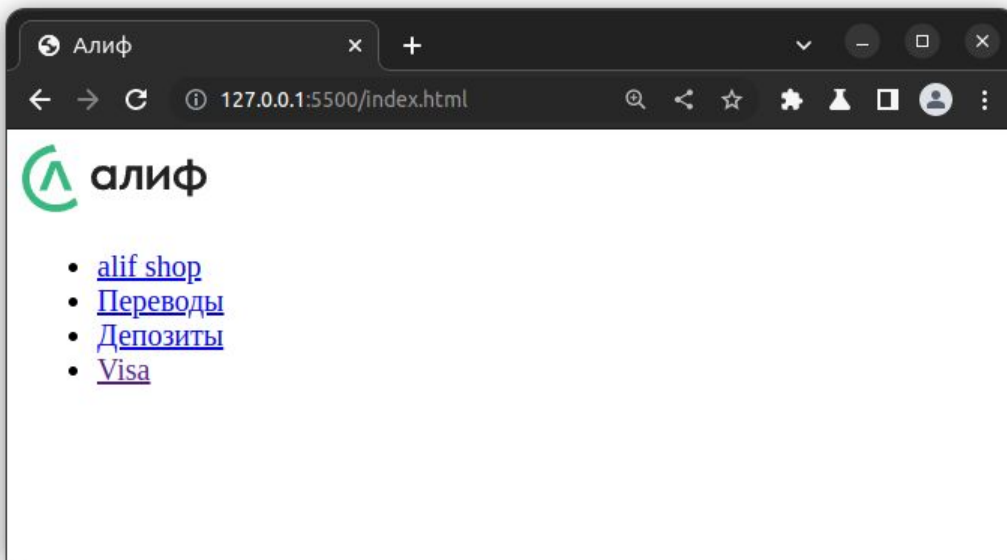
В качестве отправной точки возьмём следующий документ:

```
<> index.html > ...
1  <!DOCTYPE html>
2  <html lang="ru">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Алиф</title>
8  </head>
9
10 <body>
11     <header>
12         <nav>
13             <a href="https://alif.tj">
14                 
15             </a>
16             <ul>
17                 <li><a href="https://alifshop.tj">alif shop</a></li>
18                 <li><a href="https://intiqol.tj">Переводы</a></li>
19                 <li><a href="https://deposit.alif.tj">Депозиты</a></li>
20                 <li><a href="https://visa.alif.tj">Visa</a></li>
21             </ul>
22         </nav>
23     </header>
24 </body>
25
26 </html>
```



CSS

Если мы сейчас откроем этот документ в браузере (с помощью Live Server), то увидим, что у элементов уже есть какое-то оформление (цвета, маркеры, подчёркивание):



Давайте разбираться, откуда это всё берётся и как это поменять.



CSS

CSS (Cascading Style Sheets, каскадные таблицы стилей) – специальный язык, используемый для описания того, как должны визуально выглядеть документы (например, HTML-документы). Мы, конечно же, будем рассматривать его только в контексте создания HTML-документов (другие применения нас не интересуют).



CSS

Сам этот язык состоит из нескольких частей, описанных в разных стандартах: <https://www.w3.org/Style/CSS/current-work> и постоянно (так же, как HTML и JS) развивается.



Встраивание CSS

Для того, чтобы браузер понял, что к тому документу, который мы создали необходимо применить CSS (таблицы стилей), нам нужно как-то в самом документе указать на это (а если не укажем, он возьмёт свои, готовые, но об этом чуть позже).

Существует три варианта:

1. Элемент `style`
2. Внешний файл
3. Атрибут `style` у элемента



Элемент <style>

Рассмотрим для простоты первый вариант:

<> index.html > html > head > style

```
1  <!DOCTYPE html>
2  <html lang="ru">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Алиф</title>
8      <style>
9      |
10     </style>
11 </head>
12
13 <body>
14     <header>
15 > | <nav> ...
25 | </nav>
26 | </header>
27 </body>
28
29 </html>
```



CSS

В первом приближении (т.е. упрощённо), CSS – это набор правил, которые применяются к элементам на странице, чтобы создать визуальное оформление.

Сами правила записываются следующим образом:

```
селектор {  
    СВОЙСТВО: значение;  
    СВОЙСТВО: значение;  
}
```

```
селектор {  
    СВОЙСТВО: значение;  
    СВОЙСТВО: значение;  
}
```



Селекторы

Что такое селектор? Это специальное выражение, с помощью которого можно выбрать элемент на странице. Например, можно сказать, что все гиперссылки (тег `a` нужно раскрашивать **зелёным цветом**).



Селекторы

На текущей нашей странице:



- [alif shop](#)
 - [Переводы](#)
 - [Депозиты](#)
 - [Visa](#)
- ← ещё не переходили по этим ссылкам
- ← уже переходили (посещённая ссылка)

Т.е.:

1. Фон страницы – белый
2. Цвет текста ссылки – синий (если вы не переходили по ссылке) или фиолетовый (если уже переходили)
3. Текст ссылки подчёркнут (но картинка внутри ссылки – логотип Алиф, не подчёркивается)

Начнём знакомиться с селекторами, от самых простых к более сложным.




Селекторы

Простейший селектор – это селектор по тегу элемента, например, `a`, `img` – он позволяет выбрать все элементы с данным тегом на странице и применить к ним правило оформления (браузер сам найдёт все элементы с этим тегом и сам же применит это оформление).

Давайте попробуем:

селектор

```
a {  
  color:  green;  
}
```



свойство

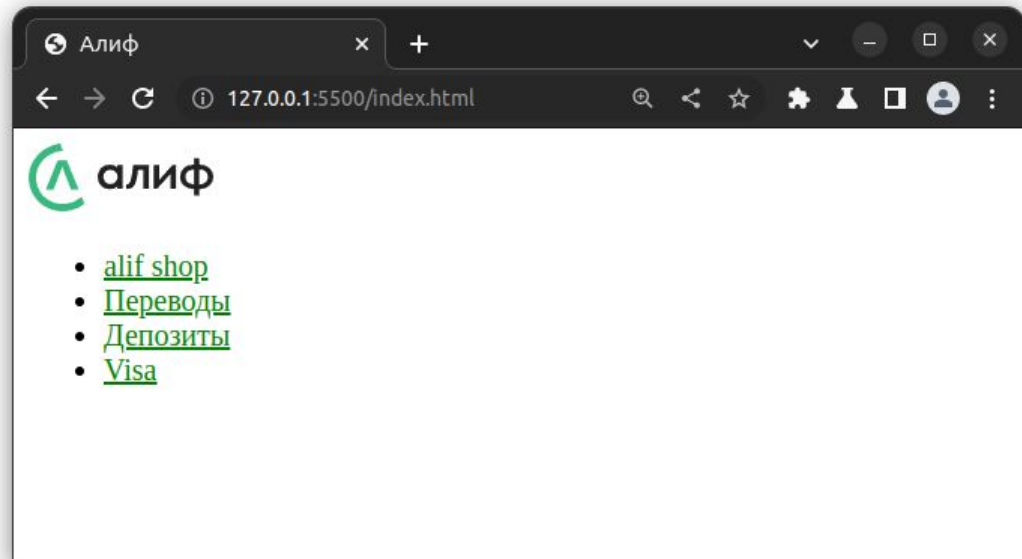
это VSCode сам показывает (писать не нужно), чтобы мы видели, каким будет цвет



Селекторы

После этого текст всех ссылок на странице станет зелёным:

```
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Алиф</title>
8   <style>
9     a {
10      color: ■green;
11    }
12  </style>
13 </head>
```



Конечно, не так зелёным, как бы нам хотелось, но давайте разбираться по-порядку.

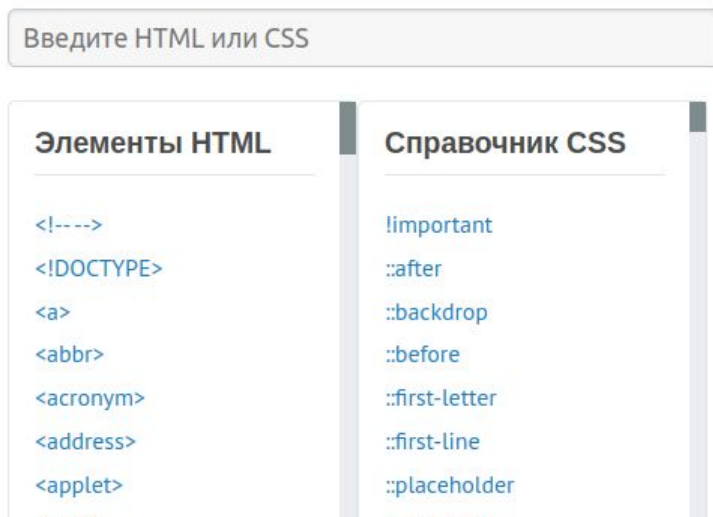


Свойства

Первый вопрос: откуда узнать, какие свойства есть и какие у них могут быть значения? В HTML у нас были элементы и их атрибуты. Все они были описаны в спецификации HTML.

С CSS всё немного сложнее: этого авторы CSS содержат специальную страницу, на которой ежегодно выкладывают актуальный документ, содержащий описания всех широко используемых свойств: <https://www.w3.org/TR/CSS/#properties>

На WebRef.ru также есть перечень свойств:



Свойства

Среди них вы можете найти то, что мы использовали - color и кликнув по нему перейти к описанию:

color

[in css-color-3](#) ← кликаем

[in css-color-3](#)

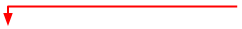
Name:	color	← название (color)
Value:	<color> inherit	← значение (либо название, либо inherit)
Initial:	depends on user agent	← начальное значение (зависит от браузера)
Applies to:	all elements	← к каким элементам применим (ко всем)
Inherited:	yes	← наследуется? (да)
Percentages:	N/A	← доступно ли указать в % от другого значения? (недоступно)
Computed value:	<ul style="list-style-type: none">The computed value for basic color keywords, RGB hex values and extended color keywords is the equivalent triplet of numerical RGB values, e.g., six digit hex value or rgb(...) functional value with an alpha value of 1.The computed value of the keyword 'transparent' is the quadruplet of all zero numerical RGBA values, e.g., rgba(0,0,0,0).For all other values, the computed value is the specified value.	← вычисление значения



Свойства

Давайте начнём по шагам: с названием понятно, это то, что мы пишем в имя свойства, т.е.:

```
a {  
  color: ...  
}
```



название



Свойства

Теперь перейдём к значению: **value** – значение в виде: `<color> | inherit`.

По порядку:

- вертикальная черта `|` обозначает, что нужно выбрать один из двух вариантов
- `<color>` (значение в угловых скобках) означает, что вместо `<color>` должно быть подставлено какое-то значение (например, `green`)
- `inherit` (значение без угловых скобок) означает, что можно написать именно это значение (прямо так и написать `inherit`)



inherit

Попробуем написать **inherit** и посмотреть, что произойдёт:

```
<style>
  a {
    color: inherit;
  }
</style>
```



- [alif shop](#)
- [Переводы](#)
- [Депозиты](#)
- [Visa](#)

стало



- [alif shop](#)
- [Переводы](#)
- [Депозиты](#)
- [Visa](#)

было

Но, если мы в Google Translator вобьём **inherit**, это не будет переводиться как чёрный:

ОПРЕДЕЛИТЬ ЯЗЫК

АНГЛИЙСКИЙ

РУССКИЙ

НЕМЕЦКИЙ



РУССКИЙ

АНГЛИЙСКИЙ

УКРАИНСКИЙ



inherit



in 'herat



7 / 5000



наследовать



nasledovat'



Наследование

Что же это значит? Это значит, что наш элемент наследует данное свойство от родителя (в терминах HTML):

```

15 <body>
16   <header>
17     <nav>
18 >   <a href="https://alif.tj">...
19
20     </a>
21   <ul>
22     <li><a href="https://alifshop.tj">alif shop</a></li>
23     <li><a href="https://intiqol.tj">Переводы</a></li>
24     <li><a href="https://deposit.alif.tj">Депозиты</a></li>
25     <li><a href="https://visa.alif.tj">Visa</a></li>

```

Т.е. **a** (для списка) наследует значение свойства **color** от **li**, тот – от **ul** и т.д.:

Name: **color**

Value: `<color>|inherit`

Initial: `depends on user agent`

Applies to: `all elements`

Inherited: **yes** ← наследуется? (да)

Percentages: `N/A`



Наследование

Это нужно запомнить: **некоторые** среди самых часто употребляемых свойств наследуются. Т.е. значение берётся у родителя, а он берёт у своего родителя и т.д.

При этом важно смотреть документацию, чтобы выяснить, наследуется или нет (например, другое свойство – [animation](#), не наследуется):

<i>Name:</i>	'animation'
<i>Value:</i>	<u><single-animation>#</u>
<i>Initial:</i>	see individual properties
<i>Applies to:</i>	<u>all elements</u>
<i>Inherited:</i>	no

не наследуется



Наследование

Стоп-стоп, но до этого у ссылки цвет был синий (или фиолетовый), а потом мы поставили зелёный (**green**). Что в этом случае случилось с наследованием?



Переопределение

Если мы сами назначаем цвет, то мы задаём своё правило (т.е. переопределяем то, что было). И тогда браузер использует наше значение:

```
<style>
  a {
    color: ■green;
  }
</style>
```

Т.е. правило очень простое*: если мы переопределяем значение, то используется наше значение, если нет, то используется то, которое приходит от родителя (если свойство наследуется).

Примечание*: прочитайте следующий слайд, чтобы понять, почему раньше ссылка была синей.



Переопределение

Но как же тогда быть с тем, что в самом начале ссылка была синей (или фиолетовой)? Ведь цвет у родителя (когда мы писали значение `inherit`) был чёрный?



User Agent

Дело в том, что иногда у HTML-документа нет собственных стилей (вспомните – до этого мы никаких стилей с вами не создавали).

Как в этом случае браузеру отображать страницу? Какие цвета использовать или размеры элементов?

Производители браузеров решили достаточно просто – они встроили в браузер собственные стили, которые всегда применяются к вашему документу.



User Agent

В спецификации браузер называют User Agent'ом и именно указание на это мы видим в описании свойства цвета:

Name:	color
Value:	<color> inherit
Initial:	depends on user agent ← зависит от браузера
Applies to:	all elements
Inherited:	yes
Percentages:	N/A

ОПРЕДЕЛИТЬ ЯЗЫК

АНГЛИЙСКИЙ

РУССКИЙ

НЕМЕЦКИЙ



РУССКИЙ

АНГЛИЙСКИЙ

УКРАИНСКИЙ



depends on user agent



21 / 5000



зависит от пользовательского агента



zavisit ot pol'zovatel'skogo agenta



Отправить отзыв



Стили браузера

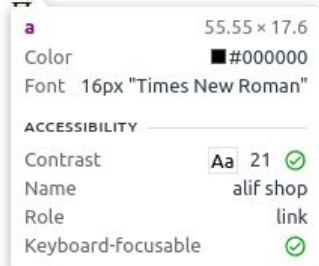
К сожалению, у каждого браузера есть свои стили. Давайте посмотрим на всю "механику" работы по применению стилей.

Для этого нажмите сочетание клавиш **Ctrl + Shift + C** и наведите указатель мыши на элемент ссылки:



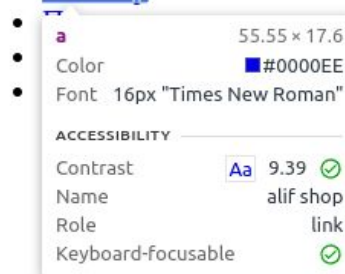
- [alif shop](#)

если значение **color: inherit**



- [alif shop](#)

если значение **color** не задано



```
<title>Алиф</title>
<style>
</style>
```



Стили браузера

А теперь кликните по ссылке – сбоку откроется панелька **Styles**:

max

приоритет

min

наши стили

уже не действует

стили браузера

```
Styles Computed Layout Event Listeners DOM Breakpoints >>
Filter :hov .cls + [icon] [icon]
element.style {
}
a {
  color: inherit;
}
a:-webkit-any-link {
  color: -webkit-link;
  cursor: pointer;
  text-decoration: underline;
}
Inherited from li
li {
  text-align: -webkit-match-parent;
}
Inherited from ul
ul {
  list-style-type: disc;
}
```

Если мы прописали стили для `a` со значением `color: inherit`, то вид будет как на изображении выше.



Стили браузера

Что это значит? Это значит, что браузер берёт свои стили, а сверху "накладывает" наши. И там, где определяются одни и те же свойства, происходит замена стилей браузера на наши значения.

При этом правила наследования всё так же действуют*:

1. Сначала накладываются наши стили на стили браузера
2. Затем выполняются расчёты в соответствии с правилами наследования

Примечание*: на самом деле, схема немного сложнее, но пока нам для запоминания будет полезно запомнить их именно в таком порядке.



Стили браузера

Зачёркивание в данном случае определяет, что наши значения переопределяют значения браузера:



The screenshot shows the Chrome DevTools 'Styles' panel. The 'Filter' bar is empty. The 'Styles' tab is selected. The 'element.style' section is collapsed. The 'a' section is expanded, showing a single rule: 'color: inherit;'. The 'a:-webkit-any-link' section is also expanded, showing three rules: 'color: -webkit-link;', 'cursor: pointer;', and 'text-decoration: underline;'. The 'a:-webkit-any-link' section is highlighted with a red vertical line. The 'a' section is highlighted with a blue vertical line. The 'a:-webkit-any-link' section is labeled 'стили браузера' (browser styles) and the 'a' section is labeled 'наши стили' (our styles).

```
element.style {  
}  
  
a {  
  color: inherit;  
}  
  
a:-webkit-any-link {  
  color: -webkit-link;  
  cursor: pointer;  
  text-decoration: underline;  
}
```

наши стили

стили браузера

А выше располагается тот, что накладывается "сверху" (в порядке наложения).



Стили браузера

Попробуем прописать следующие правила:

```
<style>
  a {
    color: ■ black;
  }
  a {
    color: ■ red;
  }
</style>
```



Это значит, что браузер читает стили сверху вниз и если правила одинаковые, то выигрывает то, которое самое нижнее.



Стили браузера

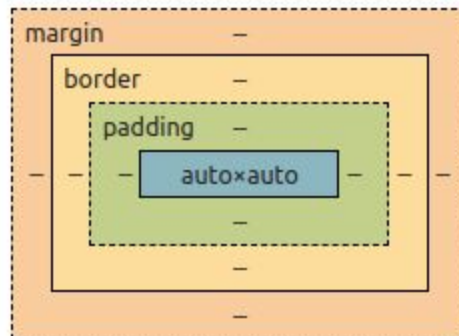
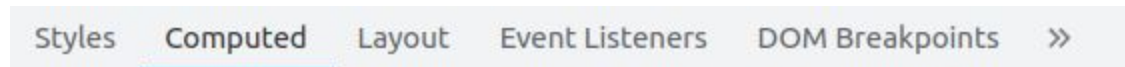
При этом браузер пишет номера строк, чтобы мы могли легко найти нужные правила:

```
8   <style>
9   |   a {
10  |     color: ■ black;
11  |   }
12  |   a {
13  |     color: ■ red;
14  |   }
15  </style>
```



Computed

Результаты расчётов вы можете увидеть на соседней вкладке **Computed**:



Filter	<input type="checkbox"/> Show all <input type="checkbox"/> Group
▶ color	■ rgb(255, 0, 0)
▶ cursor	pointer
display	inline
height	auto
▶ list-style-type	disc
▶ text-align	left
▶ text-decoration-color	■ rgb(255, 0, 0)

показывает все свойства



rgb

Q: что такое `rgb(0, 0, 0)`?

A: это специальная запись цвета. По порядку, через запятую идут значения составляющих:

- `r` – красный (red)
- `g` – зелёный (green)
- `b` – синий (blue)

Помните, мы с вами говорили про то, как числа и буквы хранятся в памяти компьютера? Так вот цвета можно хранить в виде набора трёх чисел (и смешивать их, чтобы получить нужный, как смешивают художники).



rgb

Если мы напишем в стилях просто: **red**, то получим:

► **color**





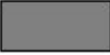
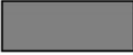


























■ `rgb(255, 0, 0)`

Но как узнать, где какой цвет? И что можно использовать? Самый простой способ – посмотреть в спецификацию (см. следующий слайд).



rgb

Color names and sRGB values

Named	Numeric	Color name	Hex rgb	Decimal
		<i>black</i>	#000000	0,0,0
		<i>silver</i>	#C0C0C0	192,192,192
		<i>gray</i>	#808080	128,128,128
		<i>white</i>	#FFFFFF	255,255,255
		<i>maroon</i>	#800000	128,0,0
		<i>red</i>	#FF0000	255,0,0
		<i>purple</i>	#800080	128,0,128
		<i>fuchsia</i>	#FF00FF	255,0,255
		<i>green</i>	#008000	0,128,0
		<i>lime</i>	#00FF00	0,255,0
		<i>olive</i>	#808000	128,128,0
		<i>yellow</i>	#FFFF00	255,255,0
		<i>navy</i>	#000080	0,0,128
		<i>blue</i>	#0000FF	0,0,255
		<i>teal</i>	#008080	0,128,128
		<i>aqua</i>	#00FFFF	0,255,255

А что такое hex-rgb? Это то же значение, но в другой форме записи:

#RRGGBB, где **RR** – это составляющая Red в шестнадцатеричной форме записи, **GG** и **BB** – Green и Blue, соответственно.

Т.е. можно писать:

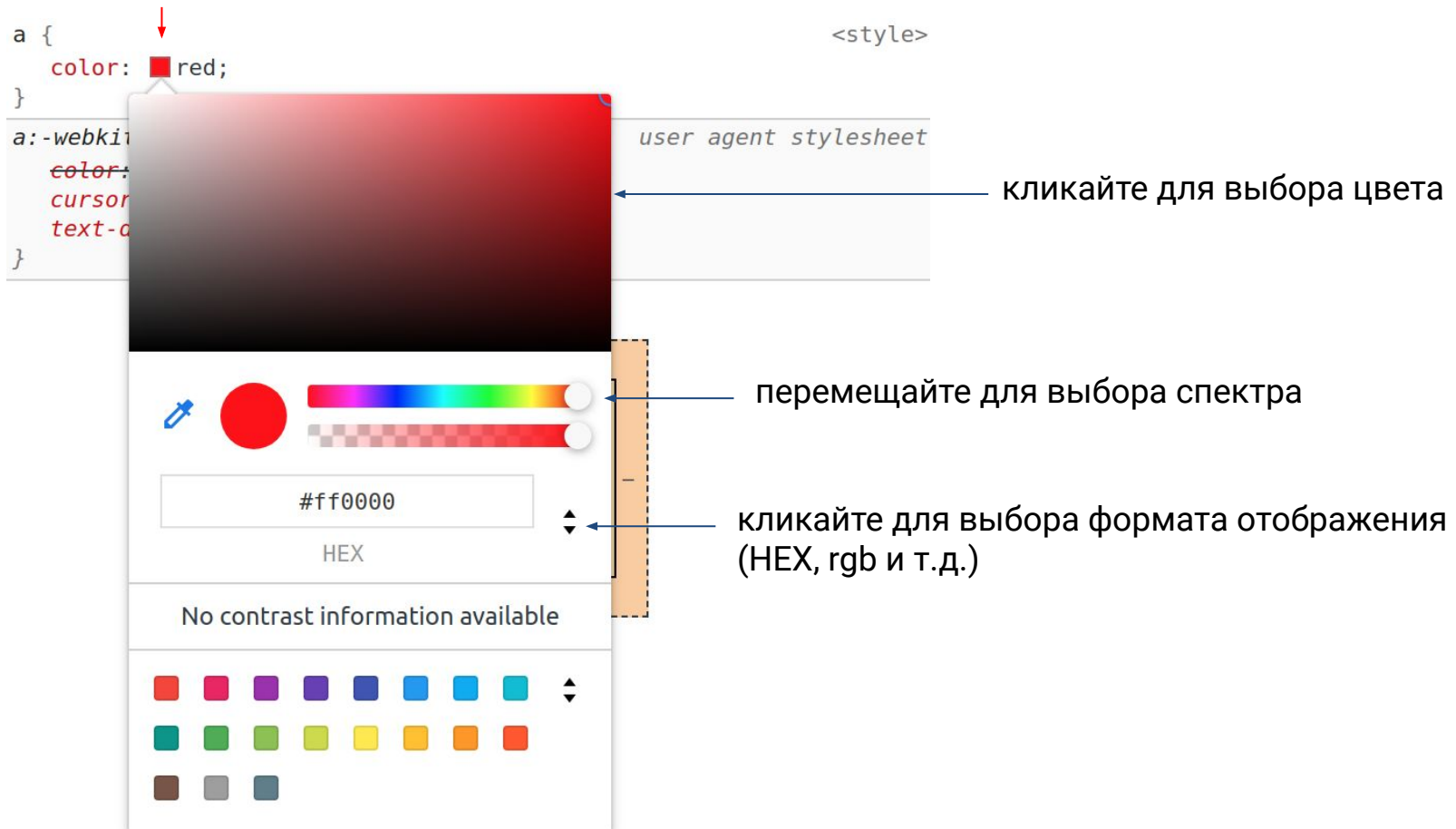
```
a {
  color: rgb(255, 0, 0);
}
a {
  color: #FF0000;
}
a {
  color: red;
}
```

Результат будет одинаковый.



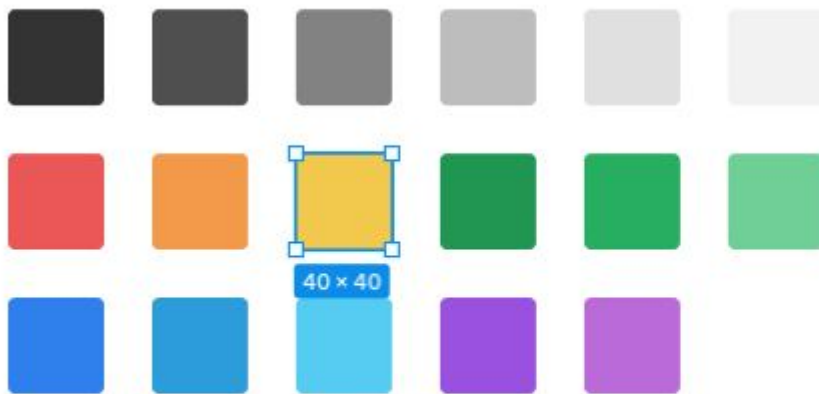
Оттенки

А если мы хотим другой цвет? Как его выбрать? Если мы установили цвет (но не `inherit`), то можно кликнуть на квадратике с цветом (такой же есть в VS Code):

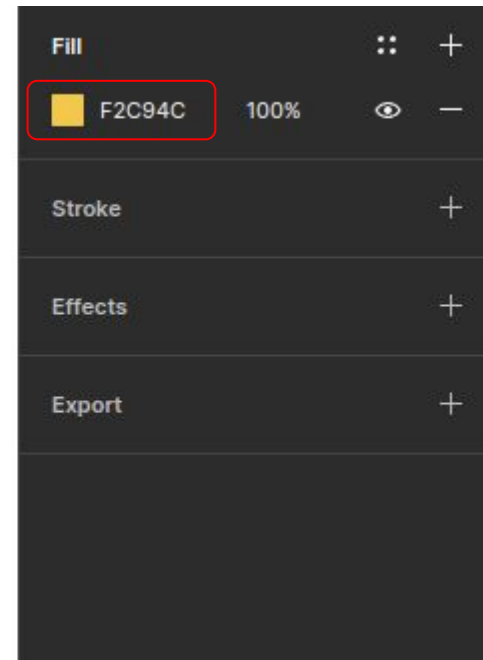


Оттенки

При работе в реальных проектах, нам чаще всего дадут файл Figma, в котором уже выставлены основные цвета (и где можно посмотреть сразу HEX-код):



This file contains your Color Styles. They are applied to the shapes in the examples above. You can use them across all your files.



Промежуточные итоги

Итак мы поговорили:

1. Про дефолтные (default – по умолчанию) стили
2. Про наследование
3. Про переопределение
4. И про важность порядка следования правил в CSS



Подключение стилей



Подключение стилей

Мы с вами говорили о том, что стили можно подключать к документу по-разному. Давайте рассмотрим все варианты и выберем самый часто используемый:

1. Элемент `style`:

```
<style>правила</style>
```

2. Атрибут `style`:

```
<a href="..." style="color: green;">alif shop</a>
```

3. Внешний файл:

link:css

</head> link:css Emmet Abbreviation

<link rel="stylesheet" href="style.css">

DOCS

- <> index.html
- # style.css



Подключение стилей

Последний вариант (с внешним файлом) является наиболее употребимым, поэтому его мы и будем далее использовать (элемент `style` и атрибут `style` стараются, по возможности, избегать, хотя мы и пройдем на следующих курсах, где и когда мы можем их встретить).



Normalize



Normalize.css

Как вы уже знаете, браузер применяет к документу свои стили (например, стиль оформления списков), которые мы можем затем уже переопределить.

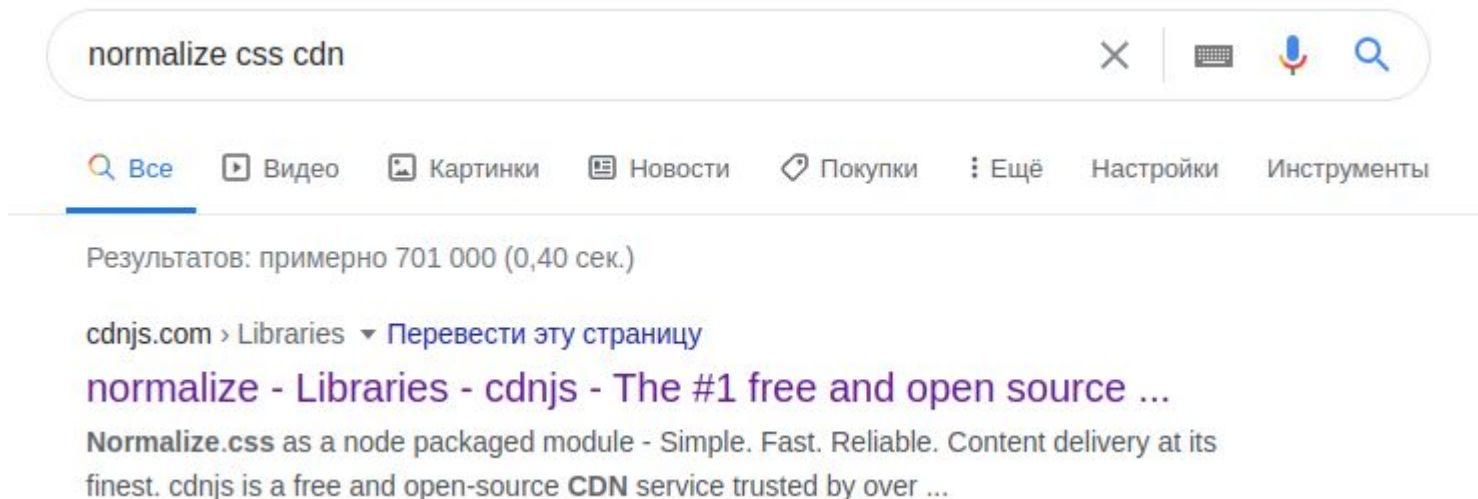
Самое ужасное – это то, что у каждого браузера это свои стили (т.е. не одинаковые). И, получается, если мы хотим, чтобы в разных браузерах веб-сайт выглядел более-менее одинаково, то придётся изучить каждый браузер и написать такие стили, которые бы "устраняли" эти недостатки.



Normalize.css

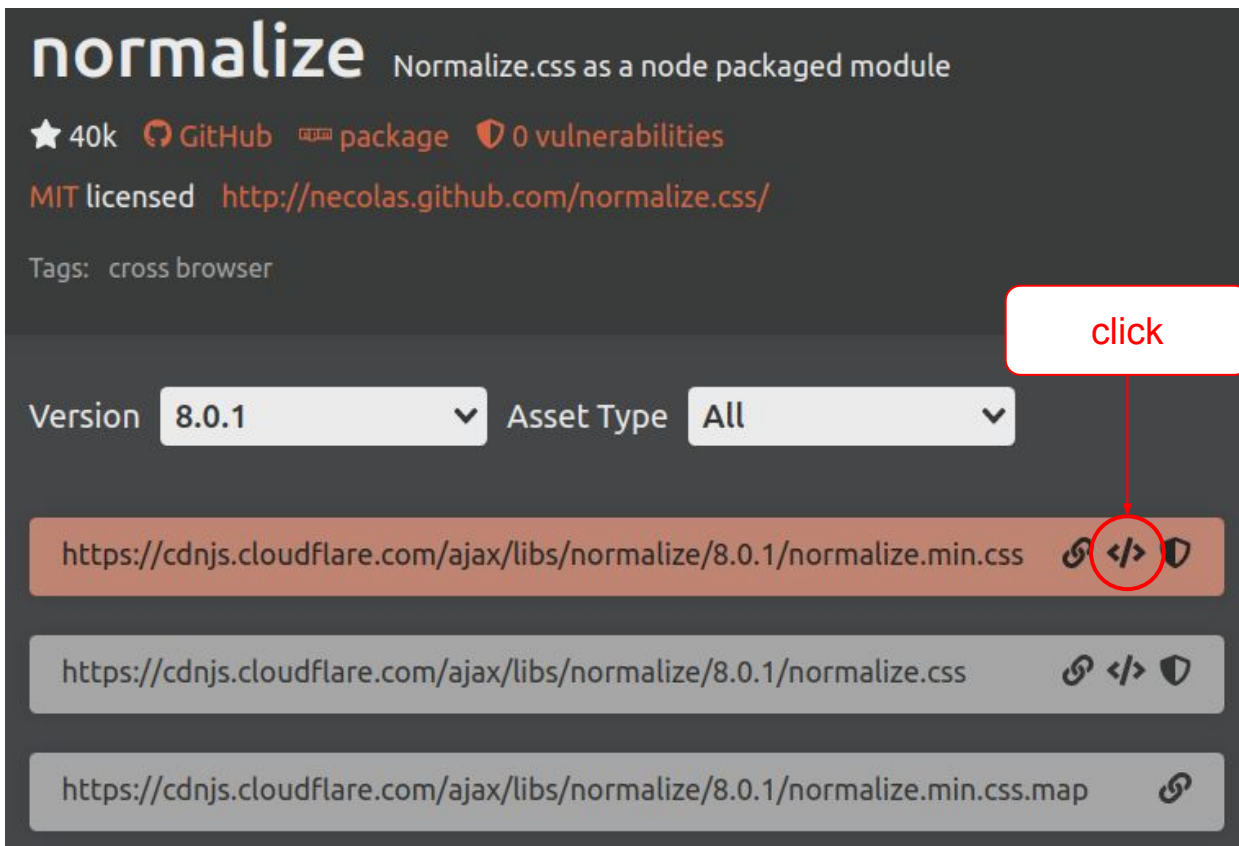
К счастью, мы с вами не одни и уже есть разработчики, которые это сделали за нас. Нам достаточно использовать их наработки (это достаточно частое явление в программировании – нет смысла каждый раз что-то делать с нуля, когда можно просто использовать готовые наработки, разумеется, если автор не против).

Как же нам их получить? Очень просто: мы в Google пишем "[normalize css cdn](#)"






CDN

Что такое CDN? CDN переводится как Content Delivery Network – сеть доставки контента. Это значит, что кто-то позаботился о том, чтобы разместить нужный нам файл в сети Интернет и нам достаточно подключить его, используя ссылку:






normalize Normalize.css as a node packaged module




★ 40k  GitHub  package  0 vulnerabilities


MIT licensed <http://necolas.github.com/normalize.css/>

Tags: cross browser

Version Asset Type

<https://cdnjs.cloudflare.com/ajax/libs/normalize/8.0.1/normalize.min.css>   

<https://cdnjs.cloudflare.com/ajax/libs/normalize/8.0.1/normalize.css>   

<https://cdnjs.cloudflare.com/ajax/libs/normalize/8.0.1/normalize.min.css.map> 



Normalize.css

А затем просто подставить его в наш документ (**Ctrl + V**) до наших собственных стилей:

```
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/normalize/8.0.1/normalize.min.css">  
<link rel="stylesheet" href="style.css">
```

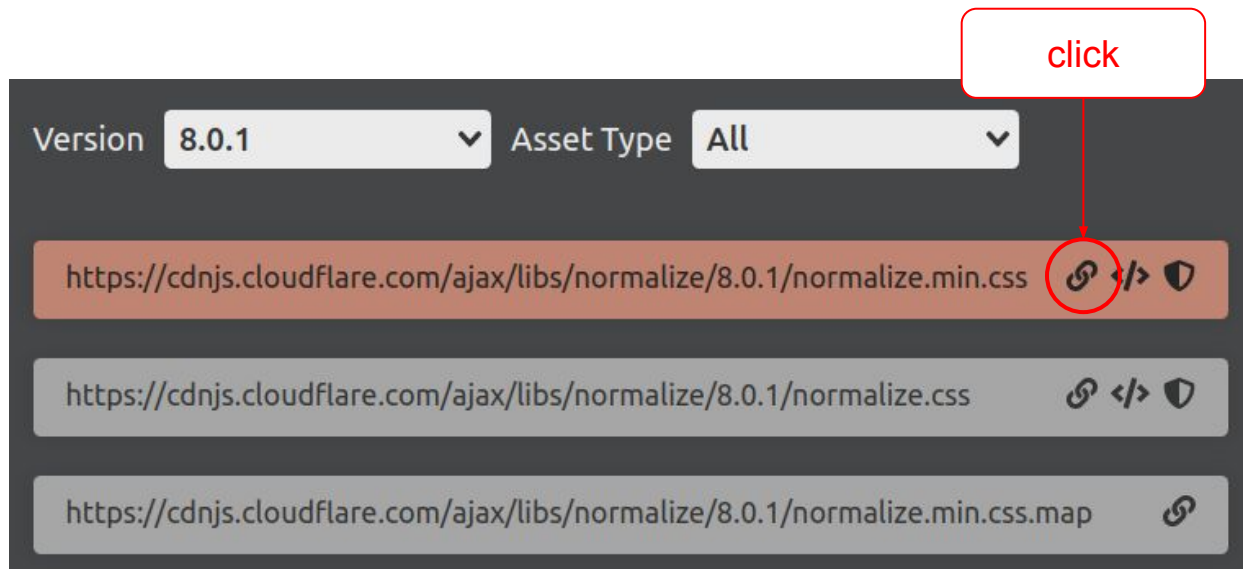
Обратите внимание, мы подставили **ДО** наших стилей, чтобы браузер сначала применил их, а потом наши (т.е. мы могли их переопределить при необходимости).



Интернет

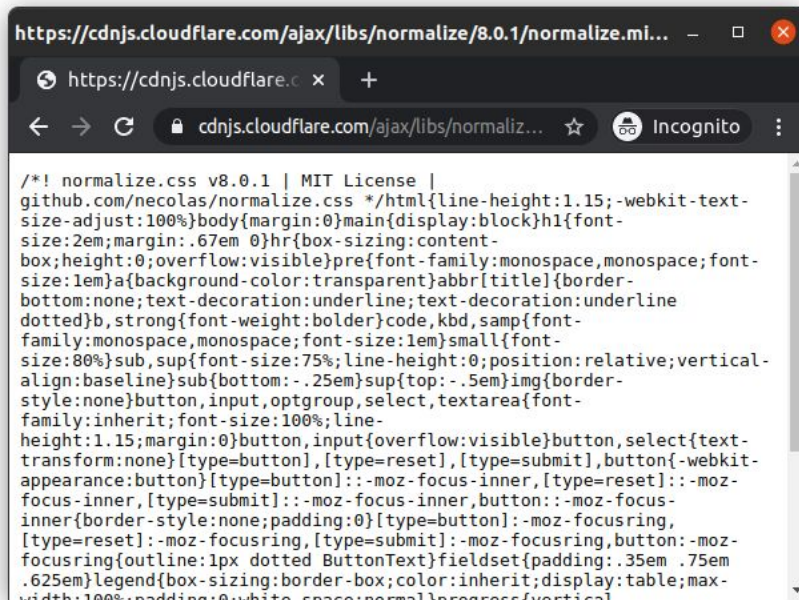
Браузер один раз загрузит из сети Интернет [normalize.css](https://cdnjs.cloudflare.com/ajax/libs/normalize/8.0.1/normalize.min.css) и будет использовать сохранённую копию.

Но если у нас есть проблемы с доступом к сети Интернет (например, он дорогой), то мы рекомендуем скачать [normalize.css](https://cdnjs.cloudflare.com/ajax/libs/normalize/8.0.1/normalize.css) в папку нашего проекта и подключать локально (тогда браузер ничего не будет скачивать):



Интернет

Подставьте скопированную ссылку в адресную строку браузера:



После чего нажмите **Ctrl + S** и выберите каталог вашего проекта.



Интернет

Должно получиться вот так:

```
▼ DOCS
  <> index.html
  # normalize.min.css
  # style.css
```

А подключать теперь нужно вот так:

```
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Алиф</title>
  <link rel="stylesheet" href="normalize.min.css">
  <link rel="stylesheet" href="style.css">
</head>
```



min

Суффикс **min** означает, что документ минимизирован (т.е. убрано всё лишнее, например, пробелы, переносы строк и т.д.).



reset

Помимо [normalize.css](#) вы будете встречать и упоминание [reset.css](#) (или его вариации) – это альтернативный подход, когда стили не приводятся к нормализованному виду, а "сбрасываются".

Не принципиально, какой из подходов вы выберите (часто это решение принимается в рамках команды на проекте), важно, что вы должны знать, что у разных браузеров разное поведение по умолчанию, поэтому нужно использовать решение, которое упростит вам жизнь и приведёт всё к единому начальному виду.



Шрифты



Шрифт

Шрифт – это начертание букв и знаков, с помощью которой отображается текст.

Определение звучит не очень, поэтому давайте сравним на картинке:

- | | |
|-----------------------------|-----------------------------|
| • alif shop | • alif shop |
| • Переводы | • Переводы |
| • Депозиты | • Депозиты |
| • Visa | • Visa |

Как вы видите, начертание (то, как отображаются буквы) разное, хотя текст один и тот же.



Шрифт

Где взять шрифт? По умолчанию, на вашем компьютере уже установлено большое количество шрифтов и можно использовать их. Но это не гарантирует, что у человека, который использует другой компьютер и другую операционную систему (Linux, Mac, Android, iOS) тоже будет установлен этот шрифт.



Шрифт

Можно взять и попросить браузер скачать нужный шрифт (как мы делали с подключением [normalize.css](#) из CDN), тогда страница будет отображаться даже тогда, когда на компьютере пользователя такого шрифта нет.

Но где же его взять? Наверное, вы уже догадались, что так же как с CSS, наверняка уже существуют сервисы, которые позволяют вам одной ссылкой подключить нужный шрифт к странице: <https://fonts.google.com>.



Google Fonts

Мы можем выбрать любой из понравившихся, главное, нужно не забыть выбрать **Cyrillic** (кириллица – набор символов для русского языка или **All Languages**):

The screenshot shows the Google Fonts website interface. At the top, there is a search bar with the text 'Search', a sentence input field with 'Sentence' and 'Type something', and a font size selector set to '40px'. Below the search bar, there are filters: 'Categories', 'Cyrillic' (selected), 'Font properties', and a checkbox for 'Show only variable fonts'. The text '118 of 1023 families' is visible on the left. On the right, there is a 'Sort by: Trending' dropdown and icons for grid and list views. The main content area displays four font family cards: 'Roboto' (12 styles), 'Open Sans' (10 styles), 'Montserrat' (18 styles), and 'Roboto Condensed' (6 styles). Each card shows the Russian text 'Алая вспышка осветила силуэт зазубренного крыла.' in the respective font. A red arrow points from a red box labeled 'click' at the bottom left to the 'Roboto' card. Another red arrow points from the 'Cyrillic' filter to the 'Roboto' card.

Search Sentence Type something 40px

Categories Cyrillic Font properties Show only variable fonts

118 of 1023 families Sort by: Trending

Roboto 12 styles
Christian Robertson

Алая вспышка осветила силуэт зазубренного крыла.

Open Sans 10 styles
Steve Matteson

Алая вспышка осветила силуэт зазубренного крыла.

Montserrat 18 styles
Julieta Ulanovsky, Sol Matas, Juan Pablo del Peral, Jacques Le Bailly

Алая вспышка осветила силуэт зазубренного крыла.

Roboto Condensed 6 styles
Christian Robertson

Алая вспышка осветила силуэт зазубренного крыла.

click



Regular 400

И на открывшейся странице выберем **Regular 400** (это значит обычный текст):

Regular 400

Принимая во внимание, что признание достоинства,

Select Regular 400 +

click

В боковой панельке появится необходимая нам код:

Use on the web

To embed a font, copy the code into the <head> of your html

```
<link rel="preconnect" href="https://  
fonts.googleapis.com">  
<link rel="preconnect" href="https://  
fonts.gstatic.com" crossorigin>  
<link href="https://fonts.googleapis.  
com/css2?family=Roboto&display=swap"  
rel="stylesheet">
```



Google Fonts

Подключим:

```
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Алиф</title>
  <link rel="stylesheet" href="normalize.min.css">
  <link rel="preconnect" href="https://fonts.googleapis.com">
  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
  <link href="https://fonts.googleapis.com/css2?family=Roboto&display=swap" rel="stylesheet">
  <link rel="stylesheet" href="style.css">
</head>
```

А в файле стилей (наших) напомним:

style.css ×

style.css > ...

```
1 body {
2   font-family: 'Roboto', sans-serif;
3 }
```

это мы взяли с той же боковой
панельки на странице шрифта

CSS rules to specify families

```
font-family: 'Roboto', sans-serif;
```

[Read our FAQs](#)



Google Fonts

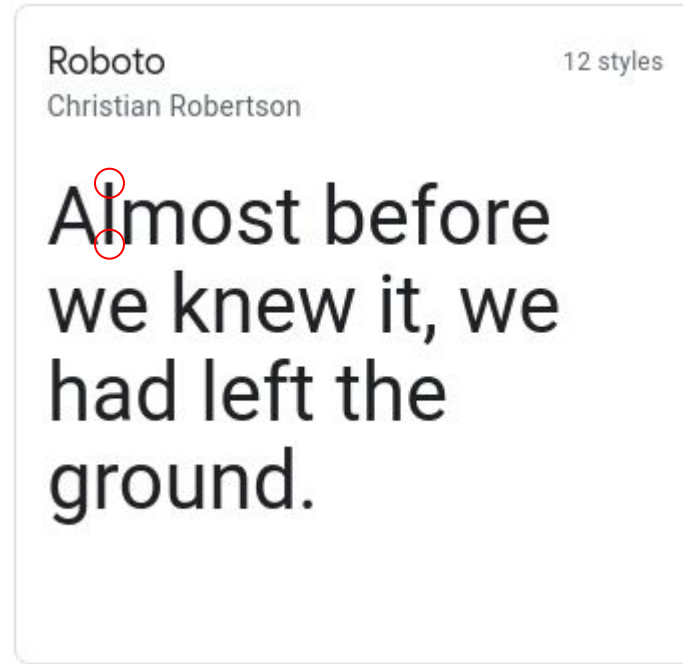
Сразу станет гораздо лучше. Но что означает эта запись через запятую?

На самом деле, в CSS мы можем указывать несколько значений одному свойству. В данном случае браузер сначала попытается загрузить шрифт с названием 'Roboto', а если не найдёт, выберет `sans-serif` (это обозначение любого шрифта без засечек).



С засечками и без засечек

Разница достаточно простая:



У шрифтов с засечками на границах букв (чаще всего верхних и нижних) есть небольшие линии (это и есть засечки).



font-family

Об этом можно прочитать на странице документации свойства [font-family](#):

'font-family'

<i>Value:</i>	<code>[[<family-name> <generic-family>] [, <family-name> <generic-family>]*] inherit</code>
<i>Initial:</i>	depends on user agent
<i>Applies to:</i>	all elements
<i>Inherited:</i>	yes
<i>Percentages:</i>	N/A
<i>Media:</i>	<u>visual</u>
<i>Computed value:</i>	as specified



Интернет

К сожалению, нет такого же простого способа подключить шрифт, скачав его из сети Интернет. Поэтому мы подготовили для вас готовый архив:

<https://alif-skills.pro/media/roboto.zip>. Вам нужно скачать его, распаковать, чтобы получилось вот так:

```
▼ DOCS
  ▼ fonts
    🚩 roboto-cyrillic-ext.woff2
    🚩 roboto-cyrillic.woff2
    🚩 roboto-greek-ext.woff2
    🚩 roboto-greek.woff2
    🚩 roboto-latin-ext.woff2
    🚩 roboto-latin.woff2
    🚩 roboto-vietnamese.woff2
    # roboto.css
  <> index.html
  # normalize.min.css
  # style.css
```



Интернет

После чего подключить вот так:

```
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Алиф</title>
8   <link rel="stylesheet" href="normalize.min.css">
9   <link rel="stylesheet" href="fonts/roboto.css">
10  <link rel="stylesheet" href="style.css">
11 </head>
```



Единицы измерения

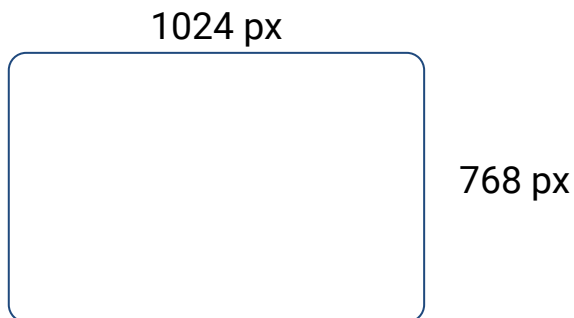


Pixel

Поговорим о единицах измерения. Единиц измерения существует достаточно большое количество, но базовой является pixel (пиксель).

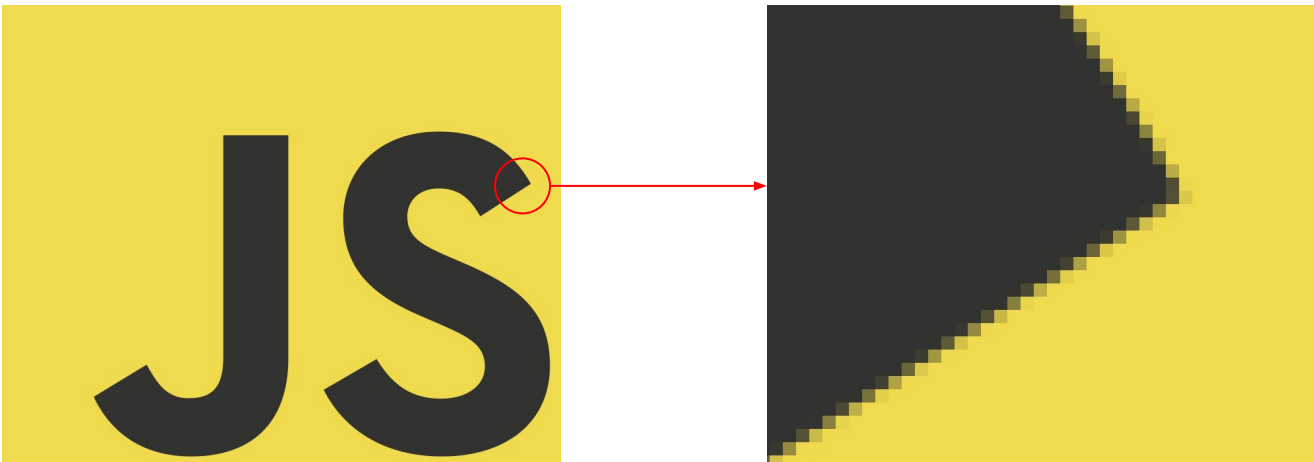
Пиксель – это одна точка на вашем мониторе. Количество таких точек по горизонтали и вертикали называют разрешением монитора и пишут, например, 1024 x 768.

Это значит, что по горизонтали у нас 1024 пиксела, а по вертикали 768:



Pixel

Пиксели специально сделаны достаточно маленькими, чтобы мы не видели их границ, но при большом приближении видно, что это всего лишь покрашенные квадратики:



Pixel

Какие бы единицы измерения мы не использовали (существуют и другие, более продвинутые единицы вроде `rem`), всё равно в конце они преобразуются в пиксели и размеры их будут отображаться в пикселях.

Поэтому мы для простоты пока будем указывать всё в пикселях.



Pixel

Зачем нам нужны были пиксели? Затем, что отталкиваясь от них, мы можем установить размер шрифта страницы (а затем и размеры других элементов).

Name:	<i>font-size</i>
Value:	<u><absolute-size></u> <u><relative-size></u> <length-percentage>
Initial:	medium
Applies to:	all elements
Inherited:	yes



font-size

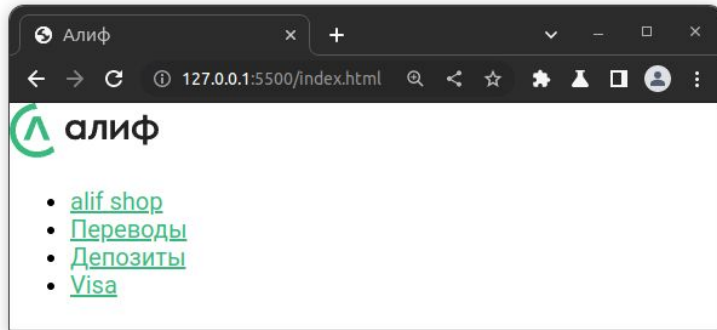
font переводится как шрифт, а **size** – размер. Таким образом, мы устанавливаем размер шрифта равным **16px** (пикселей). Можете написать другое значение (например, **160px**) и посмотреть, что изменится:

```
# style.css > ...  
1  body {  
2    font-family: 'Roboto', sans-serif;  
3    font-size: 16px;  
4  }
```



Итоги

В итоге наша страничка будет выглядеть вот так:



Пока не много, но мы только начали 😊 (а цвет ссылок вы зададите в ДЗ).



ИТОГИ



Итоги

В этой лекции мы начали изучать CSS и познакомились с основами его применения.



ДОМАШНЕЕ ЗАДАНИЕ



Орг.моменты

Практикум состоит из 8 обязательных занятий. Начиная с 23 декабря мы выкладываем новые занятия каждый Пн в 10:00 (по Душанбе).

Каждое воскресенье в 23:59 (по Душанбе) дедлайн сдачи домашнего задания.

Если не успеете сдать в срок домашнее задания, тогда этот практикум будет для вас закончен и вы сможете зарегистрироваться на запуск следующего через несколько месяцев.

Все вопросы вы сможете задавать в [Телеграм канале](#).



Основа

```

1  <!DOCTYPE html>
2  <html lang="ru">
3  <head>
4    <meta charset="UTF-8">
5    <meta name="viewport" content="width=device-width, initial-scale=1.0">
6    <title>Алиф</title>
7    <link rel="stylesheet" href="normalize.min.css">
8    <link rel="stylesheet" href="fonts/roboto.css">
9    <link rel="stylesheet" href="style.css">
10 </head>
11 <body>
12   <header>
13     <nav>
14       <a href="https://alif.tj">
15         
16       </a>
17       <ul>
18         <li><a href="https://alifshop.tj">alif shop</a></li>
19         <li><a href="https://intiqol.tj">Переводы</a></li>
20         <li><a href="https://deposit.alif.tj">Депозиты</a></li>
21         <li><a href="https://visa.alif.tj">Visa</a></li>
22       </ul>
23     </nav>
24   </header>
25   <main>
26     <h1>Обратная связь</h1>
27     <form>
28       <input placeholder="Ваше имя">
29       <textarea placeholder="Сообщение"></textarea>
30       <button>Отправить</button>
31     </form>
32   </main>
33   <footer>
34     &copy; 2022 ОАО &laquo;Алиф Банк&raquo;
35   </footer>
36 </body>
37 </html>

```

Используйте этот документ
в качестве основы.



Основа

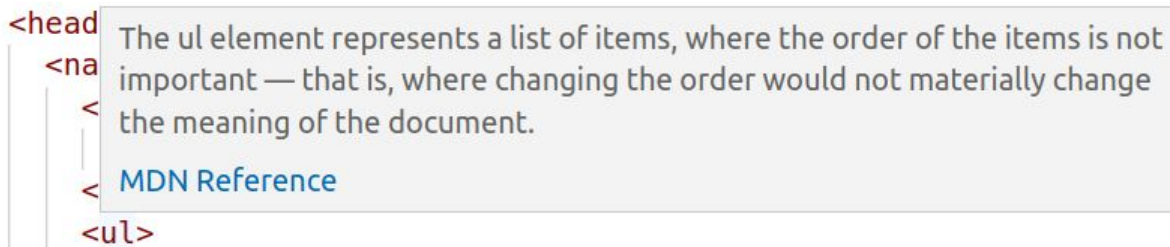
Вас не должно пугать, что некоторые теги мы не проходили. Вы самостоятельно можете найти о них информацию в спецификации HTML и на WebRef.ru



MDN

Кроме того, есть ещё один замечательный ресурс, которым мы будем на следующих курсах пользоваться чаще всего – [MDN](#) (Mozilla Developer Network). Не все статьи на нём переведены на русский язык, но те, что переведены – достаточно хорошего качества (кроме того, вы вполне можете начать потихоньку учить английский, если ещё не начали).

Кстати, именно на MDN ссылается VS Code, когда мы наводим курсор на элементы:



The screenshot shows a code editor with the following HTML code: `<head>`, `<nav>`, `<`, `<`, `<`, ``. A tooltip is displayed over the `` tag, containing the text: "The ul element represents a list of items, where the order of the items is not important — that is, where changing the order would not materially change the meaning of the document." Below this text is a link labeled "MDN Reference".



ДЗ: Фооновый цвет

Для задания фонового цвета элементов используется свойство [background-color](#).

Задайте следующий фооновый цвет элементам:

- `header` равный `#F7F8F9`
- `footer` равный `#F7F8F9`



ДЗ: Цвет ссылок

Задайте всем ссылкам на странице цвет [#39B980](#).



ДЗ: Размер заголовков

Задайте всем элементам `h1` размер в 36 пикселей.



Д3: transparent

Для цветов в CSS есть специальное ключевое слово, которое называется **transparent** (историческая справка: раньше это было ключевое слово, теперь это название цвета).

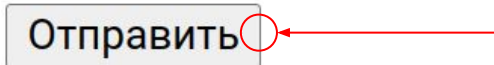
Что оно значит? Оно значит, что цвет "прозрачный" (т.е. видно то, что за элементом, если мы, например, ставим цвет фона).

Установите всем кнопкам на странице прозрачный фоновый цвет.



Д3: border-style

Вокруг кнопок есть `border` (рамка). Изучите [документацию](#) на свойство `border-style` и установите его так, чтобы рамки не было (не было совсем, ни при каких обстоятельствах).



Спасибо за внимание

alif skills

2022г.

